MDS2 Data for Project Deployment
June 19, 2003
Jennifer Schopf, Argonne National Laboratory

When deploying the MDS2, several projects have asked what is a good set of information providers to deploy along with the base installation, and what data this will deliver to the users, administrators and applications. Below is a list of attributes, and the packages that need to be installed to have this data included in an MDS2 deployment.

This document is meant to be complimentary to the MDS2 Technology Brief (http://www.globus.org/mds/mdstechnologybrief_draft4.pdf) which is meant as a basic introduction to MDS2.

Basic Resource Data

Basic cluster data will vary from project to project depending on if the architecture is solely cluster-based (like TG) or if there will also be individual machines (such as with iVDGL). If it is cluster based, and by this we mean that there are groups of machines with a queuing system front end, this data should be provided either at the cluster-level, if the clusters are homogeneous, or at the homogenous subcluster level if not. Note that this is a subset of the standard GLUE schema data. Every attribute (with the exception of very few such as names and ID's) in the GLUE schema are optional, however it has been our experience that projects want to define a minimal set of data they can expect to have published by every resource in the project.

Basic Cluster Data
- Name of cluster
- IP address
- CPU vendor, model and version
- CPU speed
- OS name, release and version
- RAM size
- Node count
- CPU count

This data is available using a standard MDS2 install along with the Ganglia provider (available from http://www.globus.org/mds/gangliaprovider.html) if your backend is running Ganglia.

File system Scratch size
- Scratch free space

This data is available using a standard MDS2 install.

Basic Queue data for each queue

Queue name
Queue policy data – max wall clock time, max total running jobs
Queue state data – number of jobs in queue, number of jobs running

This data is available using a standard MDS2 install with the GRAM reporter turned on. The GRAM reporter interfaces to PBS, LSF, Maui, etc. to pull this data.

If the project also includes individual machines, instead of queue data, you will need:

Individual Host Data
Host load

This data is data is available using a standard MDS2 install.

Service Status

In addition to basic information about the resources, projects are interested in understanding the status of the services running on those resources. Several projects have written test scripts to discover this status, most often run using a certificate user certificate as a cron job.

Basic service data includes:

Service status
Gatekeeper
Gris
Giis
Gridftp
Gsi-ssh
Condor-g
Gridpath
Mpicompile
Simple job fork job manager
Simple job pbs job manager
i job pbs job manager
Gridftp from fork job
Gridftp from pbs job
GridFTP transfer status

All of this data (except for the last, GridFTP transfer status) is attainable using Jim Basney's Scripts (http://www.ncsa.uiuc.edu/~jbasney/teragrid-setup-test.html) originally developed for the PACI resources, and then adapted for TeraGrid. We recommend setting up these scripts to run using a test-user certificate at each site (which is mapped to a "nobody" account and therefore has limited privileges) and then installing the TG Status information provider (available at http://gtr.globus.org)

The GridFP transfer status data is available from the GTR as well, and is currently configured to use the MDS service certificate to test an actual GridFTP transfer within the host.

Software information

Verifying the path and version of the software installed on a host is needed especially when working with testbeds that consist of many individual hosts. The data needed here includes:

Software information
- Software list
- Version
- Path
- Functionality (unit test)

Currently, the only information provider that supplies any of this data is the Software version and path information provider, available from the GTR (http://gtr.globus.org/article.php?story=20030424084252806) . Given a config file of software packages of interest that respond to some variation of a command line version request, this information provider advertises the version and path data of that host. Several projects (including TG) are working on test harnesses and unit tests for spcific software, but these are not yet compatible with MDS2.

Infrastructure needs

In addition to deciding what pieces of software to deploy with the standard MDS2 installation, some setup infrastructure is needed for a project.

For every project, a decision must be made about the setup of GRIS's and GIIS's. The MDS Technology Brief discusses rules of thumb for this decision.

The data from MDS2 needs to be displayed in a useful way. We currently recommend downloading the simple PHP MDS2 visualization scripts and adapting them as needed for an individual project. These are available at
http://gtr.globus.org/article.php?story=20030428204751186

Alternatively, basic web-based ldap browsers are also available. These do not need any specific tuning for different schemas in use; the one we use for the Globus GIIS (giis.globus.org) is LDAP Explorer, available from
http://igloo.its.unimelb.edu.au/LDAPExplorer

If service data is needed, the issue of certificates must be addressed. In order to gather this data, some certificate must be assigned to the agent gathering this data. In our

experience setting up a testing account mapped to the equivalent of the UNIX “nobody” account (and therefore having limited capabilities) seems to work well. If your sites do not allow this type of account, you will not be able to access this data.

Some Links to Additional Information

MDS2 website
http://www.globus.org/mds
MDS2 Technology Brief
http://www.globus.org/mds/mdstechnologybrief_draft4.pdf
Grid Technology Repository
http://gtr.globus.org
GTR information providers
http://gtr.globus.org - look at the submissions by Schopf
LDAP browser
http://igloo.its.unimelb.edu.au/LDAPExplorer
Ganglia information provider
http://www.globus.org/mds/gangliaprovider.html
PG-monitoring web page for use cases
http://www-unix.mcs.anl.gov/~schopf/pg-monitoring/